Oferta powitalna Dostęp do wszystkiego. Teraz 30% zniżki. Zaktualizuj teraz

Kodujmy przyszłość

Czytasz za darmo poprzez Friend Link serwisu Deep Concept .

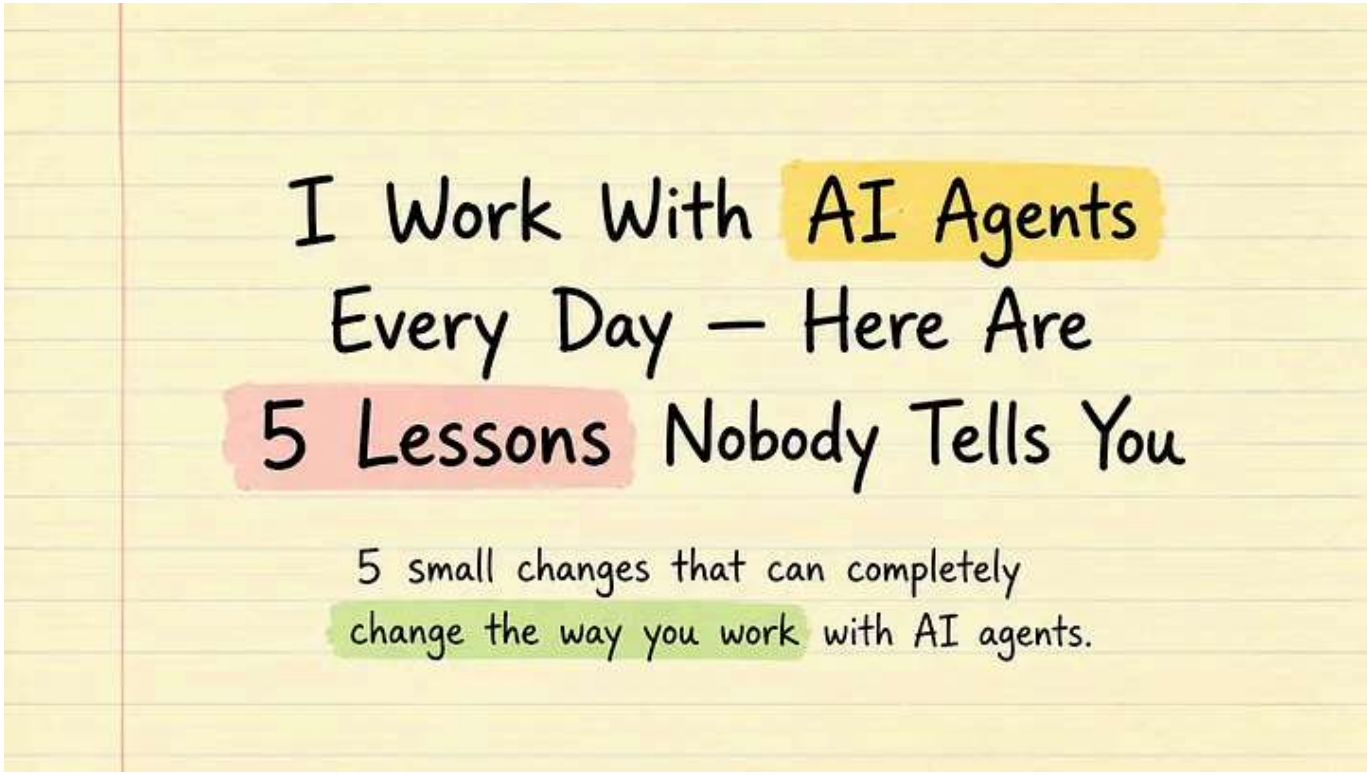
Historia dostępna tylko dla członków

Pracuję z agentami AI każdego dnia — oto 5 lekcji, których nikt ci nie powie

5 niewielkich zmian, które mogą całkowicie odmienić sposób pracy z agentami AI.

Głęboka koncepcja Następny 11 minut czytania · 16 czerwca 2026 r.

606 8 11



Hej twórcy, mam nadzieję, że mija wam dobry dzień.

> *Osoby niezrzeszone kliknij tutaj*

Dzisiaj chciałbym porozmawiać o czymś, co wydaje się bardzo realne, jeśli już korzystasz z narzędzi AI w swojej codziennej pracy.Wielu z nas pracuje już ze sztuczną inteligencją w sposób, który znacznie różni się od pracy ze zwykłymi narzędziami.

Otwieramy ChatGPT. Używamy agentów kodowania. Prosimy jedną sztuczną inteligencję o napisanie kodu, inną o jego sprawdzenie, kolejną o wyjaśnienie błędu, a czasem jeszcze inną o stworzenie dokumentacji. I nawet nie zauważając tego, nie jesteśmy już tylko „użytkownikami sztucznej inteligencji". Zarządzamy sztuczną inteligencją.

Brzmi to jak duże słowo, ale rzeczywistość jest bardzo prosta. Wydajesz instrukcje, sprawdzasz wyniki, naprawiasz błędy, zmieniasz kierunek i decydujesz, co jest wystarczająco dobre, aby iść naprzód. To w zasadzie zarządzanie.

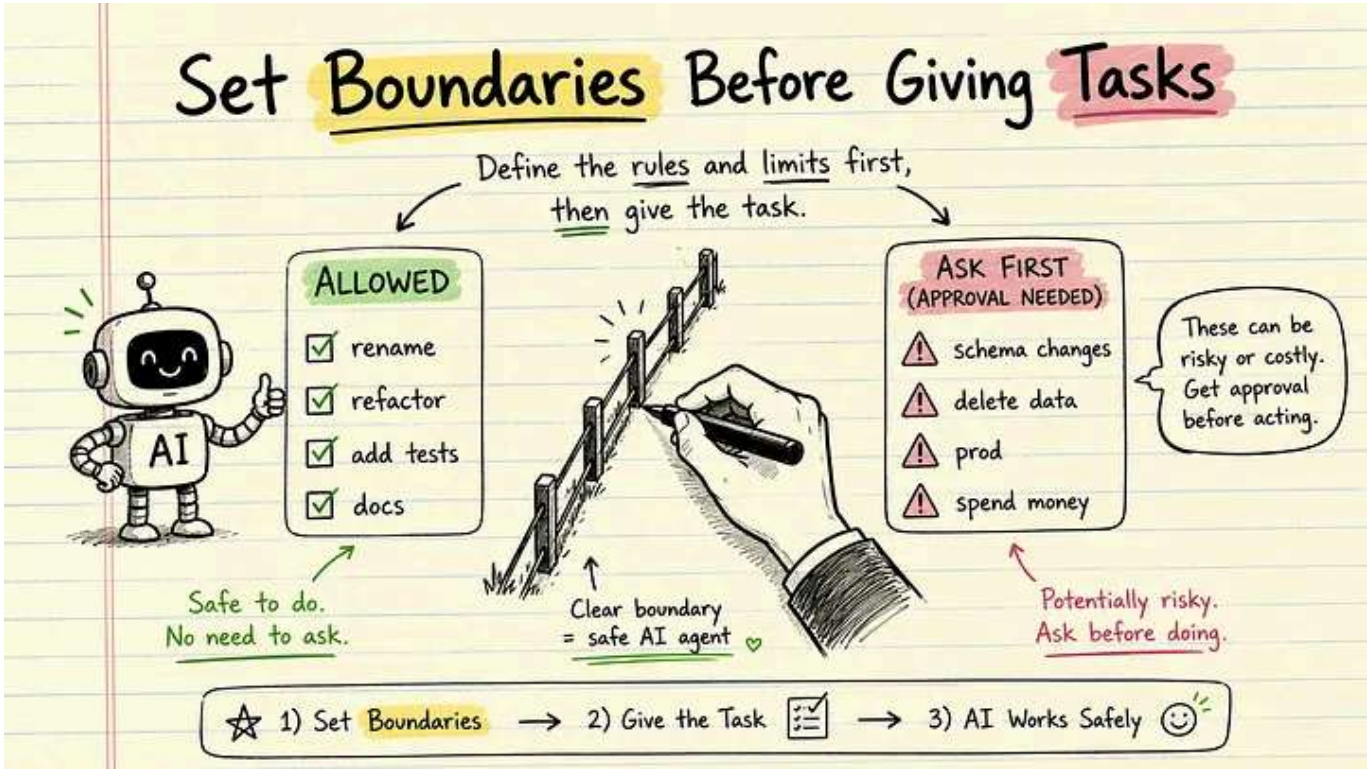
Niedawno ***PMI opublikowało również oficjalny standard dotyczący wykorzystania sztucznej inteligencji w projektach*** . To dobry krok. Jednak, jak wiele oficjalnych dokumentów, wyjaśnia on szerszy obraz niż codzienna rzeczywistość. Nie odpowiada on w pełni na proste pytanie:

Jak właściwie pracuję z agentami AI w zwykły poniedziałek, kiedy mam terminy, błędy, spotkania i prawdziwą pracę do wykonania? Ta historia nie jest więc teorią. To opowieść o moich prawdziwych doświadczeniach.

Pracuję z agentami AI niemal codziennie i nauczyłem się kilku rzeczy na własnej skórze. Czasami dawałem złe podpowiedzi. Czasami zbyt szybko

ufałem wynikom. Czasami pozwalałem agentowi zbyt długo podążać w złym kierunku. I spośród tych wszystkich drobnych błędów wywnioskowałem pięć prostych zmian, które zmieniły sposób, w jaki pracuję z AI. Nic skomplikowanego. Po prostu drobne, praktyczne zmiany, które mogą sprawić, że Twój przepływ pracy będzie bardziej przejrzysty, szybszy i mniej frustrujący.

Ustal granice przed przydzieleniem zadań



Mój pierwszy odruch związany z agentem AI był taki sam, jak w przypadku pracy z człowiekiem:

„Oto zadanie. Idź i je dokończ.”

Brzmi to normalnie, ale z agentami może to bardzo szybko stać się ryzykowne. Przekonałem się o tym, gdy agent podjął decyzję, która w danej chwili wydawała się rozsądna, ale później zdałem sobie sprawę, że niełatwo ją odwrócić. Problemem nie był agent. Problemem byłem ja.

Nigdy nie powiedziałem mu jasno, które decyzje są bezpieczne, a które wymagają mojej zgody. Dlatego teraz, zanim dam agentowi listę zadań, najpierw określam granice. Nie tylko to, co powinien zrobić. Ale także to, co może sam zdecydować. W przypadku projektów długoterminowych taki plik jest bardziej przydatny niż losowe monity, ponieważ daje agentowi jasny obszar działania.

```
# decision-boundaries.yml

 wersja: 1.0
domyślna _zasada: stop_ i _ask
agent_ tryb:
  autonomia _poziom: ograniczony
```

```
  require_ ludzki _review_ przed _merge: prawda
  require_ testy _przed_ ukończeniem: prawda
  zezwól _bezpośrednio_ na dostęp do produkcji : fałsz
dozwolony_ bez _zatwierdzenia:
 jakość  kodu :
    - ponowne formatowanie kodu
    - naprawa problemów z lintingiem
    - zmiana nazw zmiennych lokalnych w celu zapewnienia przejrzystości
    - usunięcie nieużywanych importów
    - poprawa komentarzy bez zmiany logiki
  bezpieczne _refaktoryzacja:
    - refaktoryzacja kodu wewnątrz pojedynczego modułu
    - podział dużych funkcji na mniejsze
    - poprawa struktury folderów w tym samym obszarze funkcji
    - uproszczenie zduplikowanej logiki bez zmiany zachowania
  dokumentacja:
    - aktualizacja sekcji README
    - poprawa dokumentacji wbudowanej
    - dodanie przykładów użycia
    - utworzenie notatek dla deweloperów
    - aktualizacja szkiców dziennika zmian
  testowanie:
    - dodanie testów jednostkowych
    - poprawa istniejących testów
    - dodanie brakujących testów skrajnych przypadków
    - aktualizacja nazw testów w celu zapewnienia przejrzystości
    - uruchomienie lokalnych poleceń testowych
  odwracalne_ zmiany:
    - zmiany, które można bezpiecznie cofnąć za pomocą git
    - zmiany ograniczone do plików niekrytycznych
    - zmiany, które nie wpływają na dane produkcyjne
    - zmiany, które nie modyfikują publicznych kontraktów
wymaga _zatwierdzenia:
  architektura:
    - zmiana struktury projektu w wielu modułach
    - wprowadzenie nowego frameworka
```

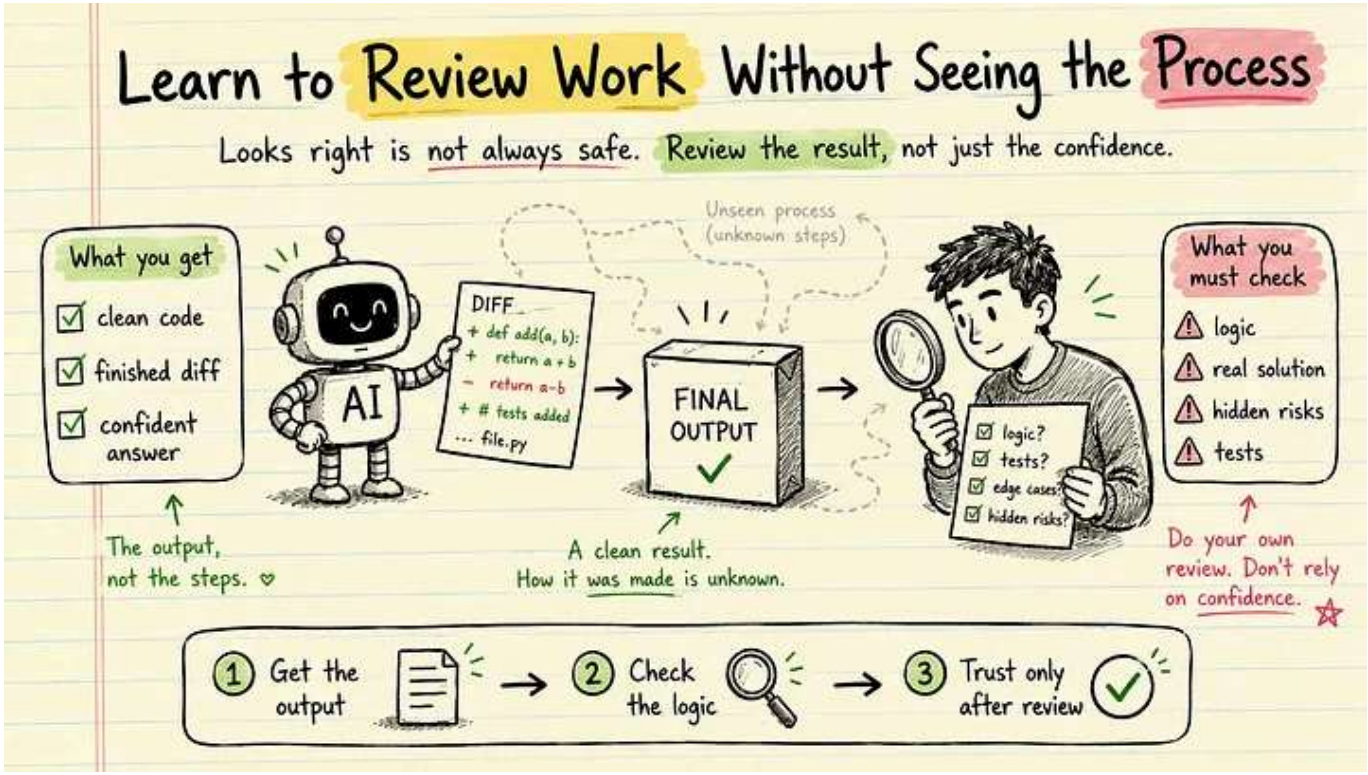
```
  - zmiana głównego przepływu aplikacji
  - zmiana logiki uwierzytelniania lub autoryzacji
  - zmiana strategii obsługi błędów
baza danych:
  - zmiany schematu
  - migracje
  - usuwanie kolumn lub tabel
  - zmiana indeksów
  - zmiana relacji między jednostkami
  - modyfikacja danych początkowych
kontrakty   API_ :
  - zmiana kształtu publicznej odpowiedzi API
  - zmiana nazw pól API
  - usuwanie pól API
  - zmiana reguł sprawdzania poprawności żądań
  - zmiana kodów statusu
  -Zmiana ładunków webhook
Bezpieczeństwo:
  - zmiany uwierzytelniania
  - zmiany autoryzacji
  - zmiany szyfrowania
  - zmiany obsługi tokenów
  - zmiany logiki haseł
  - zmiany uprawnień
  -
 Infrastruktura ustawień CORS lub plików cookie :
  - zmiany wdrażania
  - zmiany CI/CD
  - zmiany zmiennych środowiskowych
  - zmiany konfiguracji Docker lub serwera - zmiany
  zasobów w chmurze
  - monitorowanie lub rejestrowanie zmian w potoku działania
finansowe _lub_ zewnętrzne :
  - wysyłanie wiadomości e-mail lub wiadomości zewnętrznie
  - wykonywanie wywołań API, które kosztują pieniądze
  - korzystanie z płatnych usług
```

```
    - tworzenie zasobów w chmurze
    - usuwanie zasobów zewnętrznych
    - wydawanie dowolnej kwoty pieniędzy Wpływ na
  użytkownika :
    - zmiany, które wpływają na dane użytkownika
    - zmiany, które wpływają na płatności
    - zmiany, które wpływają na logowanie/rejestrację
    - zmiany, które wpływają na powiadomienia
    - zmiany, które wpływają na uprawnienia
    - zmiany, które wpływają na zachowanie środowiska produkcyjnego
Nigdy _nie_rób_ bez _jawnych_ instrukcji:
  - usuń dane produkcyjne
  - zmodyfikuj sekrety lub klucze API
  - ujawnij prywatne poświadczenia
  - wyłącz kontrole bezpieczeństwa
  - omiń testy
  - wymuś wypchnięcie do współdzielonych gałęzi
  - scal bezpośrednio z głównym
  - wdróż do środowiska produkcyjnego
  - wysyłaj prawdziwe wiadomości e-mail do użytkowników
  - wprowadzaj nieodwracalne zmiany w bazie danych Zasady _bezpieczeństwa_
 danych :   sekrety:     - nigdy nie drukuj sekretów w logach     - nigdy nie za
```

```
    - nie modyfikuj konfiguracji na żywo
    - nie wykonuj migracji bez przeglądu. Bramka
jakości :
  przed _oznaczeniem_ ukończenia:
    - wyjaśnij, co zostało zmienione
    - wyświetl listę zmodyfikowanych plików
    - wspomnij o możliwych zagrożeniach
    - potwierdź dodanie lub aktualizację testów
    - potwierdź, że żaden obszar zastrzeżony nie został naruszony
    - zasugeruj kolejny krok przeglądu,
gdy _niepewny:
  działanie: zatrzymaj_ i _zapytaj.
  Komunikat: „Nie jestem pewien, czy ta decyzja jest bezpieczna. Proszę o sprawa
```

Ten plik całkowicie zmienia relację. Wcześniej tylko przydzielałem zadania. Teraz definiuję przestrzeń decyzyjną. Agent nadal może działać szybko, ale wie, gdzie kończy się droga. Wie, co jest bezpieczne, co wymaga zatwierdzenia i czego nigdy nie należy dotykać pobieżnie. Ta drobna zmiana znacznie usprawniła mój przepływ pracy. Ponieważ prawdziwe zadanie to nie tylko powiedzieć AI, co ma robić. Prawdziwe zadanie to powiedzieć AI, gdzie nie ma iść.

Naucz się przeglądać pracę bez patrzenia na proces



Wcześniej, gdy recenzowałem czyjąś pracę, zazwyczaj miałem jakieś pojęcie o tym, jak została skonstruowana.

Znałem dyskusję. Znałem kroki. Wiedziałem, dlaczego podjęto daną decyzję.

Kiedy więc spojrzałem na końcowy wynik i „wyglądał dobrze”, mogłem mu zaufać przez większość czasu. Ale z agentami AI to się zmieniło. Czasami agent przedstawia czysty wynik końcowy, kompletną różnicę w kodzie lub

ukończony dokument, ale nie obserwowałeś całego procesu myślowego, który za nim stoi.

I tu właśnie robi się trudno. Wynik może wyglądać na czysty. Kod może wyglądać na dobrze sformatowany. Wyjaśnienie może brzmieć pewnie. Ale logika, która za nim stoi, może być nadal błędna. Dowiedziałem się, że nie mogę oceniać pracy AI w ten sam sposób, w jaki oceniam pracę zwykłych ludzi. W przypadku agentów AI „wygląda dobrze” to za mało.

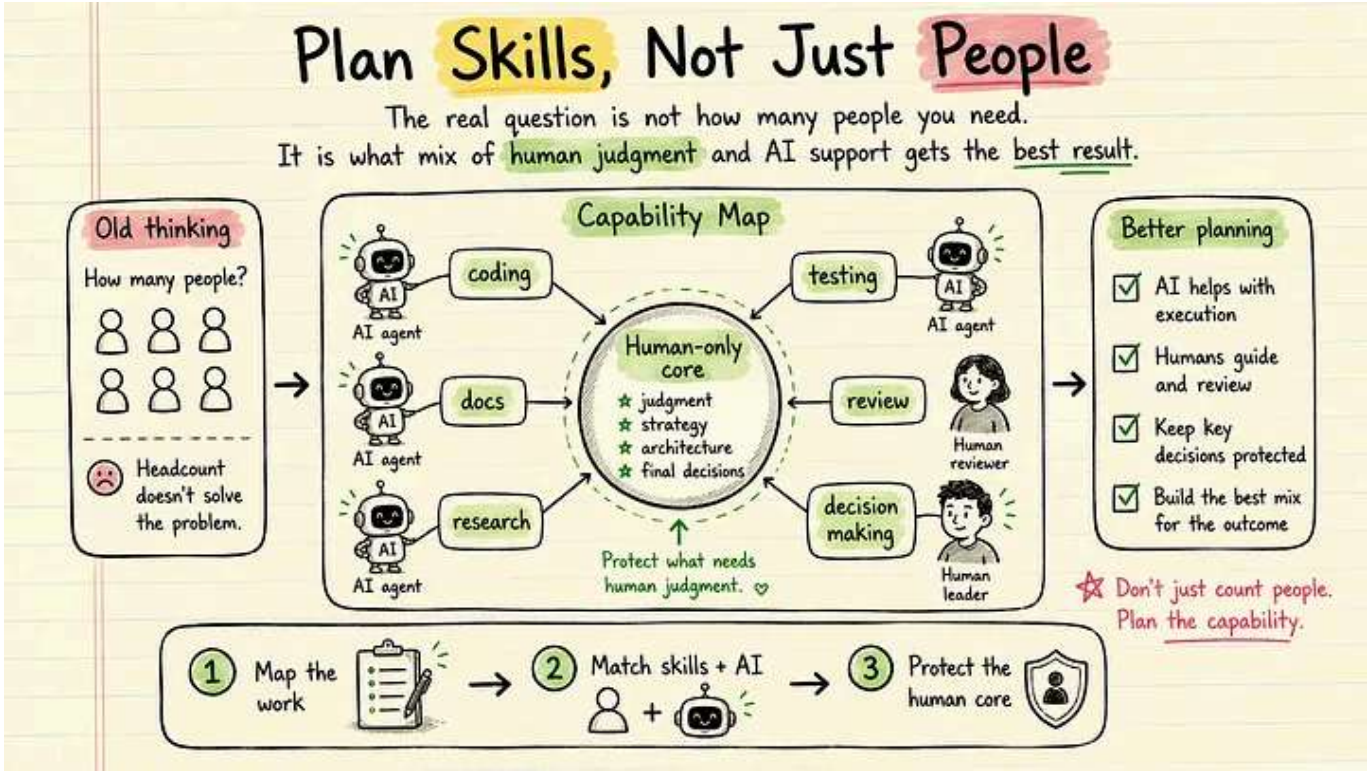
Teraz próbuję przeanalizować wynik na zimno. To znaczy, że sprawdzam wynik końcowy, nie zakładając, że proces był poprawny. Zadaję proste pytania:

1. *Czy to faktycznie rozwiązuje pierwotny problem?*
2. *Czy zmieniło coś, czego nie miało zmienić?*
3. *Czy może to uszkodzić inną część systemu?*
4. *Czy logika jest prawidłowa, czy też tylko na pozór wygląda na poprawną?*
5. *Czy istnieją testy lub dowody na to, że to działa* ?

To jedna z największych zmian w moim życiu. Moim zadaniem nie jest już pilnowanie każdego kroku. Moim zadaniem jest jasne przedstawienie zadania od samego początku, aby agent miał mniejsze szanse na pomyłkę. Bo kiedy sztuczna inteligencja potrafi pracować dłużej niż mój czas skupienia, nie mogę nadzorować każdego małego kroku.

Muszę pisać lepsze instrukcje, definiować lepsze limity i uważniej analizować efekt końcowy. Nowa umiejętność to nie tylko korzystanie ze sztucznej inteligencji. Nowa umiejętność to ocenianie wyników AI bez ślepego zaufania do niej.

Planuj umiejętności, nie tylko ludzi



Wcześniej myślałem w zupełnie normalny sposób:

„Ilu programistów potrzebujemy do tego?”

Ale po pracy z agentami AI zdałem sobie sprawę, że to już nie zawsze jest najlepsze pytanie. Lepsze pytanie brzmi:

„Jakiego rodzaju zdolności są nam potrzebne do wykonania tej pracy?”

Ponieważ teraz praca nie jest wykonywana wyłącznie przez ludzi. Niektóre części mogą być wykonywane przez agentów AI, inne wymagają ludzkiej weryfikacji, a niektóre nigdy nie powinny być w pełni przekazywane AI.

Na przykład agent AI może pomóc w pisaniu kodu, tworzeniu dokumentacji, znajdowaniu błędów, generowaniu przypadków testowych lub eksplorowaniu pomysłów.

Ale ważne decyzje wciąż wymagają zaangażowania człowieka. Kwestie takie jak kierunek rozwoju produktu, wybór architektury, wpływ na użytkownika, logika biznesowa, decyzje dotyczące bezpieczeństwa i ostateczna decyzja nie mogą być powierzane bezrefleksyjnie.

Zamiast więc planować wyłącznie liczbę pracowników, myślę teraz w kategoriach możliwości.

1. *Co sztuczna inteligencja potrafi zrobić szybko?*
2. *Co wymaga ludzkiego osądu?*
3. *Do czego potrzebne jest jedno i drugie?*
4. *Co powinno podlegać ochronie?*

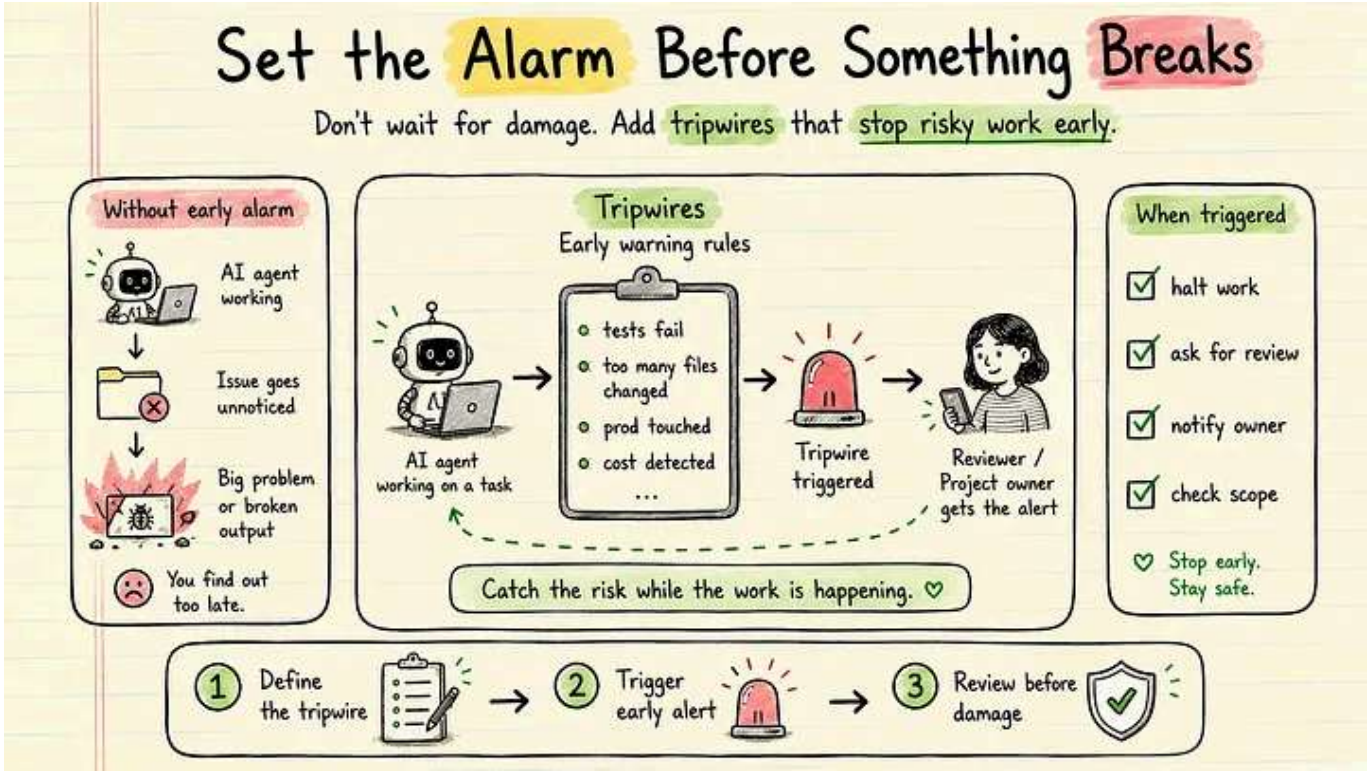
To zmieniło mój sposób myślenia o zespołach. Celem nie jest zastąpienie ludzi agentami. Celem jest zbudowanie lepiej działającego systemu, w którym zarówno ludzie, jak i agenci AI mają jasno określone role. W tym systemie najcenniejszą umiejętnością człowieka nie jest samo działanie. To osąd.

Wiedza o tym, co delegować, co analizować, co chronić i kiedy powstrzymać sztuczną inteligencję przed posunięciem się za daleko – to staje się bardzo ważną umiejętnością. Dla mnie zmiana jest prosta:

Nie pytaj tylko: „Ilu ludzi potrzebujemy?”

Zadaj sobie pytanie: „Jakie połączenie ludzi, agentów i ludzkiej oceny faktycznie przyniesie najlepszy wynik?”

Ustaw alarm zanim coś się zepsuje



W normalnej pracy zespołowej stand-upy pomagają zrozumieć, co się dzieje. Jest jednak jeden problem. Stand-up zazwyczaj informuje o tym, gdy coś już poszło nie tak. W przypadku agentów AI czekanie do końca jest ryzykowne.

Bo czasami agent może działać bardzo szybko, ale w złym kierunku. Może zmienić zbyt wiele plików, dotknąć niewłaściwego obszaru lub po cichu zniszczyć coś, co na początku wydawało się małe.

Więc teraz staram się ustawić alarm, zanim wybuchnie pożar. Używam małego `tripwires.yml` pliku. Pomysł jest prosty:

Jeżeli coś przekroczy bezpieczną granicę, przerwij pracę i powiadom mnie.

```yaml
# tripwires.yml
default_action: halt_and_ask_for_review
tripwires:
  - name: tests_failed
    watch: test_pass_rate
    trip: "< 100% na zmienionych plikach"
    action: halt_and_notify_owner
    owner: human_reviewer
    reason: "Nowy lub zmodyfikowany kod nie może być kontynuowany z nieudanymi
  - name: too_many_files_changed
    watch: files_changed
    trip: "> 20 plików w jednym zadaniu"
    action: pause_for_scope_review
    owner: project_owner
    reason: "Duże zmiany mogą ukrywać nieoczekiwane efekty uboczne."
  - name: protected_files_touched
    watch: touched_files
    trip:
      includes:
        - ".env"
        - "auth/*"
        - "payments/*"
        - "database/migrations/*"
        - "production-config/*"
    action: halt_and_request_approval
    owner: tech_lead
```

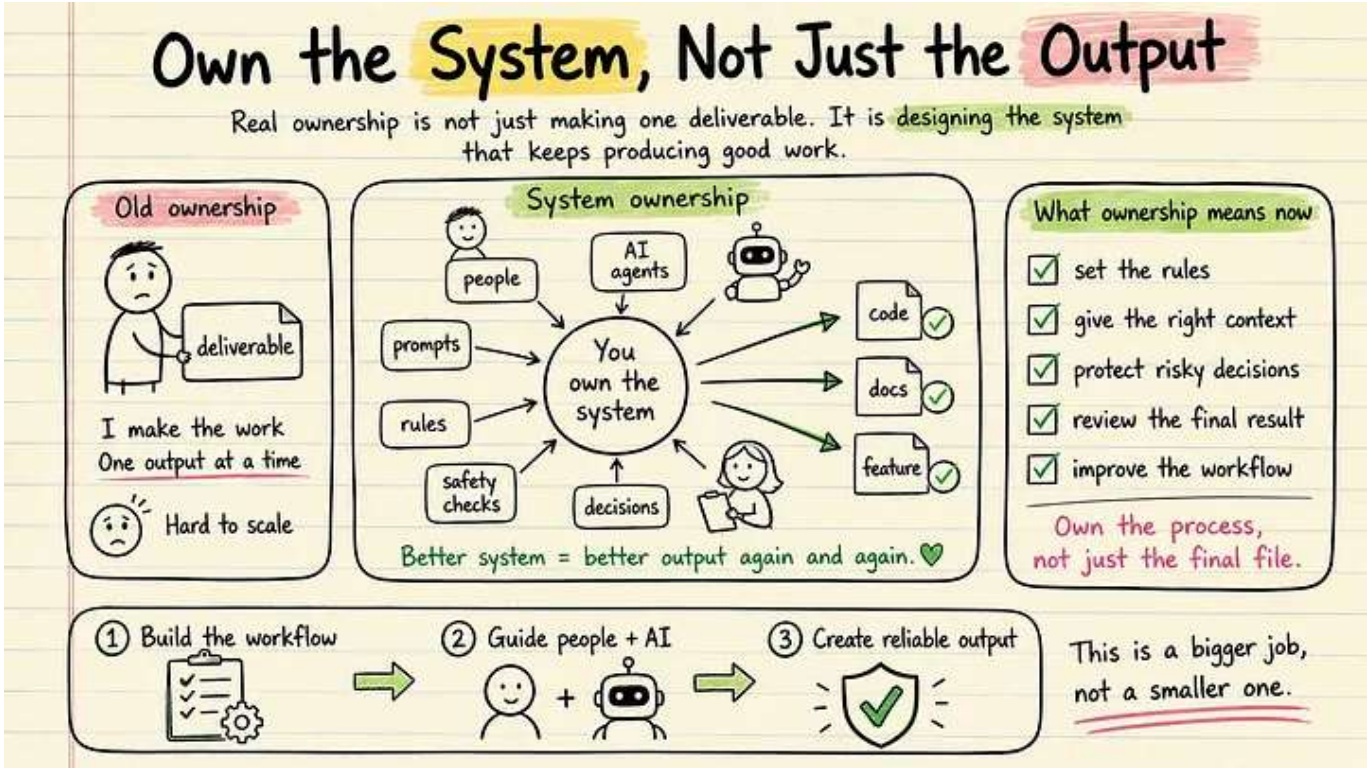
```
  reason:  "Zabezpieczone pliki wymagają zatwierdzenia przez człowieka przed k
- nazwa:  public_api_changed
  obserwacja:  api_contract
  wycieczka:  "request_or_response_shape_changed"
  akcja:  halt_and_request_approval
  właściciel:  backend_owner
  powód:  "Zmiany w kontrakcie API mogą spowodować awarię front-endu, aplikacj
- nazwa:  database_change_detected
  obserwacja:  database_schema
  wycieczka:  "schema_or_migration_changed"
  akcja:  halt_and_request_approval
  właściciel:  database_owner
  powód:  "Zmiany w bazie danych mogą być trudne do odwrócenia."
- nazwa:  external_action_detected
  obserwacja:  external_side_effects
  wycieczka:
    obejmuje:
      - "email_send"
      - "payment_action"
      - "cloud_resource_create"
      - "third_party_api_write"
  akcja:  halt_and_request_approval
  właściciel:  project_owner
  powód:  "Wszystko, co ma wpływ na świat zewnętrzny, wymaga przeglądu."
- nazwa:  wykryty_koszt
  zegarek:  szacowany_koszt
  podróż:  "> 0"
  działanie:  zatrzymanie_i_żądanie_zatwierdzenia
  właściciel: budget_owner
  reason:  „Agent nie powinien wydawać pieniędzy bez zgody”
. nazwa:  security_sensitive_change
  obserwacja:  security_area
  wycieczka:
    zawiera:
      - „uwierzytelnianie”
      - „autoryzacja”
```

```
        - „tokeny”
        - „pliki cookie”
        - „CORS”
        - „szyfrowanie”
      akcja: halt_and_request_approval
      właściciel: security_reviewer
      powód: „Zmiany zabezpieczeń muszą zostać dokładnie sprawdzone”.
    - nazwa: unreaquirement
      obserwacja: zaufanie
      wycieczka: „niski_lub_niepewny”
      akcja: stop_and_ask
      właściciel: task_owner
      powód: „Agent nie powinien zgadywać, gdy wymaganie jest niejasne”.
  raportowanie:
    on_trip:
      zawiera:
        - tripwire_name
        - powód
        - files_affected
        - suggested_next_step
        - risk_level
    message_template: |
      Wyzwalany Tripwire: {tripwire_name}
      Powód: {reason}
      Pliki dotknięte: {files_affected}
      Sugerowany następny krok: {suggested_next_step}
```

Wygląda to bardzo prosto, ale oszczędza sporo stresu. Celem nie jest bowiem zatrzymanie pracy agenta. Chodzi o to, aby zatrzymać go, zanim stworzy większy problem. Wcześniej sprawdzałem uszkodzenia po wykonaniu

zadania. Teraz staram się wychwycić ryzyko, gdy zadanie jest jeszcze w toku. Ta jedna mała zmiana bardzo zmieniła mój tok pracy. Ponieważ w przypadku agentów AI najlepszy przegląd nie zawsze jest na końcu. Czasami najlepszym przeglądem jest alarm, który wcześnie zatrzymuje niewłaściwą pracę.

Bądź właścicielem systemu, a nie tylko wyników



To dla mnie największa zmiana. Wcześniej poczucie własności było proste.

Kod był mój. Dokument był mój. Funkcja była moja. Wynik był mój. I szczerze mówiąc, to nadal jest prawdą. Ale z agentami AI poziom odpowiedzialności się zmienia. Teraz nie jestem właścicielem tylko ostatecznego produktu. Jestem właścicielem systemu, który go tworzy. Ten system obejmuje ludzi, agentów AI, monity, proces przeglądu, zasady, kontrole bezpieczeństwa i ostateczny proces decyzyjny.

To zupełnie inny rodzaj odpowiedzialności. Ponieważ kiedy agent AI pisze kod, tworzy dokument, naprawia błąd lub generuje pomysły, nie mogę po prostu powiedzieć:

„To zrobiła sztuczna inteligencja”.

NIE.

Wynik nadal jest moją odpowiedzialnością. Ale moje zadanie nie polega już tylko na samodzielnym tworzeniu każdego małego elementu. Moim zadaniem jest zaprojektowanie przepływu pracy, w którym dobra praca

będzie mogła być produkowana wielokrotnie. To oznacza, że muszę zadawać lepsze pytania:

1. *Czy agent otrzymuje właściwy kontekst?*
2. *Czy granice są jasne?*
3. *Czy istnieje etap przeglądu?*
4. *Czy ryzykowne decyzje są chronione?*
5. *Czy końcowy wynik rzeczywiście rozwiązuje rzeczywisty problem?*

To nie jest mniejsze zadanie. To większe. Ponieważ stworzenie jednego produktu jest przydatne. Ale stworzenie systemu, który może generować wiele dobrych produktów, to punkt wyjścia prawdziwej skali. To właśnie awans daje nam sztuczna inteligencja. Przechodzimy od ręcznego wykonywania każdego zadania do posiadania systemu, który wykonuje pracę.

W tym systemie ludzki osąd staje się jeszcze ważniejszy. Agent może tworzyć. Ale nadal musisz kierować, analizować, chronić i podejmować decyzje. Ostatnia lekcja jest więc prosta:

Twoim zadaniem nie jest tylko posiadanie dzieła. Twoim zadaniem jest posiadanie systemu, który za nim stoi.

A więc, jakie jest Twoje stanowisko?

Oto pięć zmian, których nauczyłem się z własnego doświadczenia, głównie dzięki popełnianiu błędów. Nadal nie jestem w nich idealny.

Czasem dobrze definiuję granice. Czasem zlecam sztucznej inteligencji chaotyczne zadanie i liczę, że jakimś cudem wszystko zrozumie. Czasem dokładnie analizuję. Czasem łapię się na tym, że zbyt szybko ufam czystym wynikom. Oto więc proste pytanie dla Ciebie:

Oceń siebie w skali od 1 do 5 po każdej zmianie.

1 oznacza „Rzadko to robię”.
5 oznacza „To już jest częścią mojego przepływu pracy”.

Dwa obszary, w których uzyskałeś najniższy wynik, to prawdopodobnie dwa zachowania, które powinieneś poprawić w tym kwartale. Nie zmieniając wszystkiego naraz. Wystarczy wprowadzić jedną, niewielką zmianę w

sposobie pracy z agentami AI. Dla mnie pierwszą dużą zmianą było ustalenie granic przed przydzieleniem zadań.

To, nad czym wciąż pracuję, to przeglądanie wyników AI bez ślepego zaufania do nich tylko dlatego, że wyglądają na czyste. To jest prawdziwa krzywa uczenia się. Nie poznawanie każdego nowego narzędzia AI. Uczenie się, jak pracować z AI w sposób bezpieczny, przejrzysty i rzeczywiście użyteczny.

Jeśli ta historia Ci pomogła, podziel się nią z innymi programistami, członkami zespołu lub przyjaciółmi, którzy również chcą dowiedzieć się, jak lepiej współpracować z agentami AI.

I chciałbym wiedzieć:

Którą z tych pięciu zmian już wykonujesz, a której jeszcze unikasz?

> ***Uwaga redaktora*** *: Ta historia opiera się na mojej własnej wiedzy, doświadczeniu i codziennej pracy z narzędziami i agentami AI. Narzędzia AI, przepływy pracy i*

> *najlepsze praktyki zmieniają się bardzo szybko, dlatego przed ich zastosowaniem należy je zweryfikować pod kątem własnego projektu, zespołu i przypadku użycia. Ta historia zawiera również treści wspomagane przez AI. Wykorzystałem AI do usprawnienia komunikacji, ustrukturyzowania instrukcji i stworzenia treści wizualnych dla lepszego zrozumienia. Jednak główne idee, wnioski i perspektywa, którymi się tu dzielę, pochodzą z mojego własnego doświadczenia.*

Sztuczna inteligencja | Sztuczna inteligencja | Programowanie | Rozwój oprogramowania | Technologia

Opublikowano w Let's Code Future

12,7 tys. obserwujących · Ostatnia publikacja 17 godzin temu

Następny

Witamy w Let's Code Future! 🚀 Dzielimy się historiami na temat rozwoju oprogramowania, sztucznej inteligencji, produktywności, samodoskonalenia i przywództwa, aby pomóc Ci się rozwijać, wprowadzać innowacje i wyprzedzać konkurencję. Dołącz do nas i pomóż nam kształtować przyszłość – historia po historii!

Napisane przez Deep concept

5 tys. obserwujących · 482 obserwujących

Następny

Odpowiedzi (8)

Arturdob

What are your thoughts?

Qixotic Software
Jun 16

Are these yaml files linked in an AGENTS file and passed into context each time?

61 3 replies Reply

Syed M Aqil Burney
Jun 16

Nicely written your heuristic way of working and domain awareness ,other way get advantage from your experience but heterogeneity might crate problems and it would be of course on the part of the one who applies your precious experience ,but any way a good way to learn from others experience .thanks.

54 2 replies Reply

Nryady he
3 days ago

> Many of us are already working with AI in a way that feels very different from normal tools.

It genuinely feels different because the dynamic has shifted from using a tool to managing a relationship.

5 Reply

See all responses

Pomoc Status O Kariera Naciskać Blog Sklep Prywatność Zasady Warunki Tekst na mowę